

[illegible]

[illegible]



```
1 0001 0 MODULE setclustr (
2 0002 0 IDENT= 'V04-000'
3 0003 0 LANGUAGE (BLISS32),
4 0004 0 ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL=LONG_RELATIVE)
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1 ++
31 0031 1
32 0032 1 FACILITY: SET CLUSTER
33 0033 1
34 0034 1 ABSTRACT: Implement SET CLUSTER qualifiers
35 0035 1
36 0036 1 AUTHOR: Paul R. Beck
37 0037 1
38 0038 1 DATE: 19-FEB-1984
39 0039 1
40 0040 1 REVISION HISTORY:
41 0041 1
42 0042 1 V03-001 DAS0001 David Solomon 09-Jul-1984
43 0043 1 Fix truncation errors; make nonexternal refs LONG_RELATIVE.
44 0044 1
45 0045 1 --
```

```

47 0046 1 LIBRARY 'SYSS$LIBRARY:LIB';
48 0047 1 LIBRARY 'SYSS$LIBRARY:CLIMAC.L32';      ! CLI macros
49 0048 1
50 0049 1 External routines
51 0050 1
52 0051 1 EXTERNAL ROUTINE
53 0052 1     set$validate_quorum,      ! Validate and set new quorum value
54 0053 1     cli$present,             ! Get qualifiers
55 0054 1     cli$get_value,           ! Get values of qualifiers
56 0055 1     lib$put_output,          ! $PUT message to SYSS$OUTPUT
57 0056 1     lib$cvdt_dtb;            ! Convert ASCII to numerical
58 0057 1
59 0058 1 Literal data definitions
60 0059 1
61 0060 1 LITERAL
62 0061 1     true = 1;
63 0062 1     false = 0;
64 0063 1
65 0064 1
66 0065 1 Declare the error messages
67 0066 1
68 0067 1 Definitions in [CLIUTL.SRC]SET.MSG
69 0068 1
70 0069 1 EXTERNAL LITERAL
71 0070 1     set$_nocluster,           ! node not in cluster
72 0071 1     set$_syntax;             ! syntax error
73 0072 1
74 0073 1 Routines used in this module
75 0074 1
76 0075 1 FORWARD ROUTINE
77 0076 1     set$cluster;              ! Analyze qualifiers for SET CLUSTER
78 0077 1
79 0078 1 Macros to generate a string with a descriptor.
80 0079 1
81 0080 1 MACRO
82 M 0081 1     ASCID[]=%CHARCOUNT(%REMAINING), ! string descriptor
83 0082 1     UPLIT BYTE(%REMAINING)%;
```



```

85 0083 1 %SBTTL 'SET$CLUSTER'
86 0084 1 ++
87 0085 1 SET$CLUSTER
88 0086 1
89 0087 1 FUNCTIONAL DESCRIPTION:
90 0088 1
91 0089 1     Add support for the following commands:
92 0090 1         SET CLUSTER/QUORUM=n      ! set active quorum to specified value
93 0091 1     Further description of each qualifier accompanies the code implementing
94 0092 1     that qualifier.
95 0093 1
96 0094 1 CALLING SEQUENCE:
97 0095 1
98 0096 1     CALL SET$CLUSTER
99 0097 1
100 0098 1 FORMAL PARAMETERS:
101 0099 1
102 0100 1     none
103 0101 1
104 0102 1 IMPLICIT INPUTS:
105 0103 1
106 0104 1     Qualifiers are obtained through calls to the CLI.
107 0105 1
108 0106 1 IMPLICIT OUTPUTS:
109 0107 1
110 0108 1     none
111 0109 1
112 0110 1 ROUTINE VALUE:
113 0111 1
114 0112 1     none
115 0113 1
116 0114 1 COMPLETION CODES:
117 0115 1
118 0116 1     none
119 0117 1
120 0118 1 SIDE EFFECTS:
121 0119 1
122 0120 1 --
123 0121 1
124 0122 1 GLOBAL ROUTINE set$cluster =
125 0123 2 BEGIN
126 0124 2
127 0125 2 LOCAL
128 0126 2     oldpriv      : $BBLOCK [8],          ! Permanent privileges
129 0127 2     desc       : $BBLOCK [dsc$c_s_bln], ! general descriptor
130 0128 2     quorum     : LONG,                  ! /QUORUM=q value
131 0129 2     new_quorum  : LONG,                  ! quorum value set
132 0130 2     fao_buffer  : BLOCK [132,BYTE],      ! buffer used with FAO
133 0131 2     fao_desc    : $BBLOCK [dsc$c_s_bln], ! descriptor for same
134 0132 2     output_desc : $BBLOCK [dsc$c_s_bln], ! work descriptor for output
135 0133 2     status      : LONG;                  ! completion status
136 0134 2
137 0135 2 OWN
138 0136 2     quorum_ctrstr : VECTOR [2,LONG]          ! FAO control string
139 0137 2     INITIAL ( ASCID('%SET-I-QUORUM, new quorum is !UW.'));
140 0138 2
141 0139 2 EXTERNAL
```

```

: 142      0140 2      clu$gl_club;                                ! =0 if no cluster
: 143      0141 2      :
: 144      0142 2      Verify that caller has sufficient privileges. SYSPRV, OPER and CMKRNL
: 145      0143 2      are required.
: 146      0144 2      :
: 147      0145 2      IF NOT (status = $SETPRV (ENBFLG = 1,          ! Enable
: 148      0146 2      PRVADR = 0,                                ! No new privileges
: 149      0147 2      PRMFLG = 1,                                ! Permanent privs
: 150      0148 2      PRVPRV = o(dpriv))                          ! Store current ones here
: 151      0149 2      THEN SIGNAL_STOP (.status);
: 152      0150 2      :
: 153      0151 2      Check to see if privileges there. If not, return error.
: 154      0152 2      :
: 155      0153 2      IF NOT .oldpriv [prv$v_oper] THEN status = $$$_NOOPER;
: 156      0154 2      IF NOT .status
: 157      0155 2      THEN
: 158      0156 2      BEGIN
: 159      0157 2      SIGNAL (.status);
: 160      0158 2      RETURN false;
: 161      0159 2      END;
: 162      0160 2      :
: 163      0161 2      If this node is not in a cluster, there's not much point in proceeding.
: 164      0162 2      :
: 165      0163 2      IF .clu$gl_club EQL 0
: 166      0164 2      THEN
: 167      0165 2      BEGIN
: 168      0166 2      SIGNAL (set$_nocluster);
: 169      0167 2      RETURN false;
: 170      0168 2      END;

```



/QUORUM = q

```
172 0169 2 %SBTTL ' /QUORUM = q'
173 0170
174 0171
175 0172 Initialize dynamic descriptors
176 0173
177 0174 $init_dyndesc (desc);
178 0175 $init_dyndesc (fao_desc);
179 0176 $init_dyndesc (output_desc);
180 0177 fao_desc [dsc$w_length] = 132;
181 0178 fao_desc [dsc$a_pointer] = fao_buffer;
182 0179 output_desc [dsc$w_length] = 0;
183 0180 output_desc [dsc$a_pointer] = fao_buffer;
184 0181
185 0182 status = cli$present ( %ASCII 'QUORUM' );
186 0183 IF .status
187 0184 THEN
188 0185 BEGIN
189 0186
190 0187 SET CLUSTER /QUORUM = n
191 0188 The value supplied will be used as a new active quorum value across
192 0189 the entire cluster. It is checked against high and low limits:
193 0190  $V \leq Q \leq (2V+2)/2$  ! V = current active votes
194 0191 and is pinned to the high or low limit as appropriate. If the value
195 0192 supplied is 0, the "proper" quorum for the current cluster is
196 0193 supplied:  $(2V+2)/2$ 
197 0194
198 0195 IF NOT cli$get_value ( %ASCII 'QUORUM', desc )
199 0196 THEN
200 0197 quorum = 0
201 0198 ELSE
202 0199 IF (NOT lib$cvdt_dtb ( .desc [dsc$w_length],
203 0200 .desc [dsc$a_pointer],
204 0201 quorum ))
205 0202 OR (.quorum GTR 65535)
206 0203 THEN
207 0204 BEGIN
208 0205
209 0206 QUORUM must be a number no larger than a word.
210 0207
211 0208 SIGNAL(set$syntax, 1, desc);
212 0209 RETURN false;
213 0210 END;
214 0211
215 0212 The rest of the work must be done synchronized with the connection
216 0213 manager.
217 0214
218 0215 status = set$validate_quorum (.quorum, new_quorum);
219 0216 IF (.status) AND (.new_quorum NEQ 0)
220 0217 THEN
221 0218
222 0219 Report the quorum value which was set
223 0220
224 0221 IF $FAO ( quorum_ctrstr, ctrstr
225 0222 output_desc [dsc$w_length], outlen
226 0223 fao_desc, outbuf
227 0224 .new_quorum ) p1 = quorum
228 0225 THEN
```

0122  
0148  
0149  
0153  
0154  
0157  
0163



		0C	12	00043	BNEQ	5\$		
		8F	DD	00045	PUSHL	#SETS_NOCLUSTER		0166
63	00000000G	01	FB	0004B	CALLS	#1, LIB\$SIGNAL		
		00C8	31	0004E	BRW	10\$		0167
F0	AD 020E0000	8F	D0	00051	MOVL	#34471936, DESC		0174
		F4	AD	00059	CLRL	DESC+4		
10	AE 020E0000	8F	D0	0005C	MOVL	#34471936, FAO_DESC		0175
		14	AE	D4	00064	CLRL	FAO_DESC+4	
08	AE 020E0000	8F	D0	00067	MOVL	#34471936, OUTPUT_DESC		0176
		0C	AE	D4	0006F	CLRL	OUTPUT_DESC+4	
10	AE	84	8F	9B	MOVZBW	#132, FAO_DESC		0177
14	AE	18	AE	9E	MOVAB	FAO_BUFFER, FAO_DESC+4		0178
		08	AE	B4	CLRW	OUTPUT_DESC		0179
0C	AE	18	AE	9E	MOVAB	FAO_BUFFER, OUTPUT_DESC+4		0180
		00000000'	EF	9F	PUSHAB	P.AAB		0182
00000000G	00		01	FB	CALLS	#1, CLISPRESNT		
	52		50	D0	MOVL	R0, STATUS		
	7E		52	E9	BLBC	STATUS, 9\$		0183
		F0	AD	9F	PUSHAB	DESC		0195
		00000000'	EF	9F	PUSHAB	P.AAD		
00000000G	00		02	FB	CALLS	#2, CLISGET_VALUE		
	04		50	E8	BLBS	R0, 6\$		
			6E	D4	CLRL	QUORUM		0197
			2C	11	BRB	8\$		
			5E	DD	PUSHL	SP		0199
		F4	AD	DD	PUSHL	DESC+4		0200
	7E	F0	AD	3C	MOVZWL	DESC, -(SP)		0199
00000000G	00		03	FB	CALLS	#3, LIB\$CVT_DTB		
	09		50	E9	BLBC	R0, 7\$		
0000FFFF	8F		6E	D1	CPL	QUORUM, #65535		0202
			10	15	BLEQ	8\$		
		F0	AD	9F	PUSHAB	DESC		0208
			01	DD	PUSHL	#1		
		00000000G	8F	DD	PUSHL	#SETS_SYNTAX		
		63	03	FB	CALLS	#3, LIB\$SIGNAL		
			3F	11	BRB	10\$		0209
		04	AE	9F	PUSHAB	NEW_QUORUM		0215
		04	AE	DD	PUSHL	QUORUM		
00000000G	00		02	FB	CALLS	#2, SET\$VALIDATE_QUORUM		
	52		50	D0	MOVL	R0, STATUS		
	28		52	E9	BLBC	STATUS, 9\$		0216
		04	AE	D5	TSTL	NEW_QUORUM		
			23	13	BEQL	9\$		
		04	AE	DD	PUSHL	NEW_QUORUM		0224
		14	AE	9F	PUSHAB	FAO_DESC		
		10	AE	9F	PUSHAB	OUTPUT_DESC		
		00000000'	EF	9F	PUSHAB	QUORUM_CTRSTR		
00000000G	00		04	FB	CALLS	#4, SY\$FAO		
	0A		50	E9	BLBC	R0, 9\$		
		08	AE	9F	PUSHAB	OUTPUT_DESC		0226
00000000G	00		01	FB	CALLS	#1, LIB\$PUT_OUTPUT		
	50		01	D0	MOVL	#1, R0		0231
			04	00118	RET			
			50	D4	CLRL	R0		0232
			04	0011B	RET			

; Routine Size: 284 bytes, Routine Base: \$CODE\$ + 0000

SETCLUSTER  
V04-000

/QUORUM = q

D 9  
16-Sep-1984 00:42:29  
14-Sep-1984 12:09:00

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]SETCLUSTER.B32;1

Page 8  
(4)



SETCLUSTER  
V04-000

/QUORUM = q

E 9  
16-Sep-1984 00:42:29  
14-Sep-1984 12:09:00

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]SETCLUSTER.B32;1

Page 9  
(5)

: 237 0233 1 END  
: 238 0234 0 ELUDOM

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
SPLITS	68	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
SOWNS	8	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
SCODES	284	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
-\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	14	0	1000	00:01.9
-\$255\$DUA28:[SYSLIB]CLIMAC.L32;1	14	0	0	9	00:00.1

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:SETCLUSTER/OBJ=OBJ\$:SETCLUSTER MSRC\$:SETCLUSTER/UPDATE=(ENH\$:SETCLUSTER)

: Size: 284 code + 76 data bytes  
: Run Time: 00:07.6  
: Elapsed Time: 00:29.5  
: Lines/CPU Min: 1852  
: Lexemes/CPU-Min: 11715  
: Memory Used: 103 pages  
: Compilation Complete



0052 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

